Laboratorio No. 4 Plataforma base

# Objetivo

* Instalar y configurar software base

# Herramientas a utilizar

* Computadores
* Acceso a Internet

# Introducción Informe

En este laboratorio vamos a conocer unos comandos útiles al momento de conocer el estado del computador y los procesos que corren en él, también, a conocer cómo se pueden automatizar tareas y verificar el estado de la red y sus distintas características en la VM. Por último, vamos a ver un poco más a fondo el funcionamiento de los distintos motores de base de datos (MariaDB y PostgresSQL) y cómo podemos conectarnos a base de datos remota desde distintas maquinas virtuales.

# Marco Teorico

**Crontab** : Cron es el nombre del programa (demonio o servicio que sólo necesita ejecutarse en el arranque del sistema) que permite a usuarios Linux/Unix ejecutar automáticamente comandos o scripts (grupos de comandos) a una hora o fecha específica. Es usado normalmente para comandos de tareas administrativas, como respaldos, pero puede ser usado para ejecutar cualquier cosa.

**Ps**: Para mostrar una instantánea de los procesos actuales que se están ejecutando y su estado se utiliza el comando ps (process status). Este comando muestra una línea de cabecera seguida de una serie de líneas de información con los procesos del usuario actual, que se ejecutan en terminales. La información se ordena por terminal y la ID (PID) del proceso que se está ejecutando.

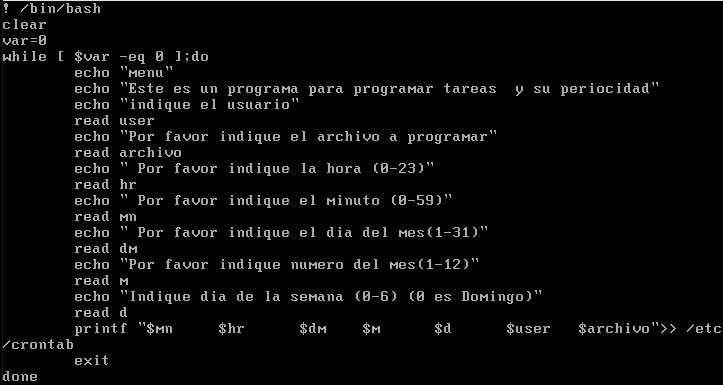
**procesos del sistema:** Codigos ejecutables que están en todo el sistema operativo en ejecución. Un proceso en el OS viene identificado por un número único denominado ID o PID y por el identificador del proceso que lo ha creado llamado PPID o proceso padre.

**Ps:** Este comando se ejecuta normalmente en lo que se suele llamar una terminal inteligente, lo que le permite estar mostrando la información actualizada en la terminal de forma ininterrumpida hasta que ejecutemos el comando ‘q’ para finalizar el análisis. Se utilizarán tantas líneas de información como líneas disponibles tenga la terminal, aunque está cantidad se puede variar a nuestro gusto por medio de sus opciones.

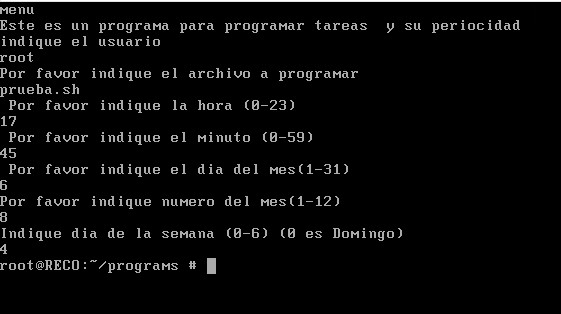
## Otros comandos útiles

* 1. Escriba un programa e Shell para los servidores Linux Slackware y FreeBSD que permita configurar una tarea que se ejecute periódicamente en el sistema. El usuario indicará la tarea que se desea ejecutar y la periodicidad del mismo.

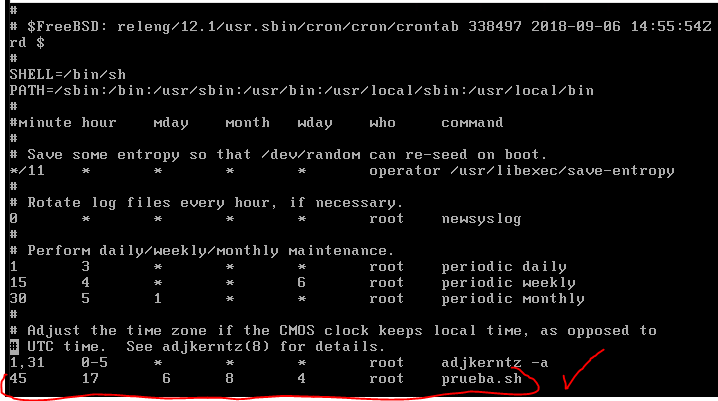
Primero creamos un Shell en donde se le soliciten los datos al usuario, luego de ello con los datos proporcionados se añade como una nueva línea de texto al archivo etc /crontab



Ingresamos los datos solicitados:

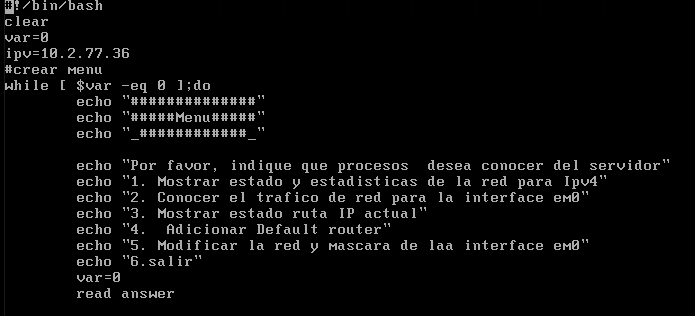


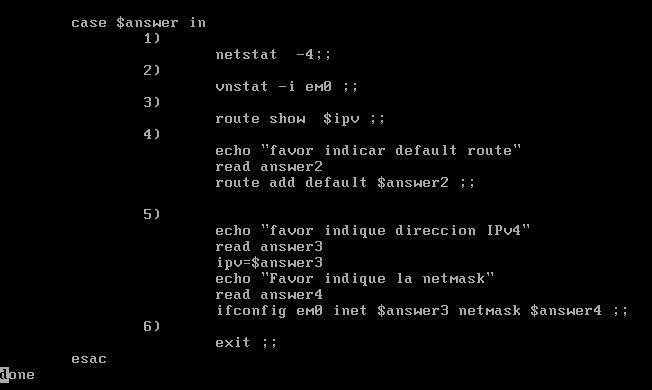
Luego verificamos el archivo y vemos que se ejecutó correctamente



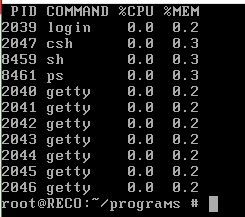
* 1. Estudie el funcionamiento de comandos que dan información de la red, ej: netstat, vnstat, route y ethtool (o equivalentes en Slackware y FreeBSD) revise diferentes parámetros que se puedan usar y cree un programa en Shell que los use (cree un menú con al menos 5 diferentes opciones que muestres diferentes ejecuciones de los comandos). **Los estudiantes deben ser capaces de entender la salida de la ejecución de los comandos y presentarla de forma fácil de entender**.

Primero creamos un menú interactivo en donde le pedimos a los usuarios que indiquen el tipo de actividad a realizar para que pruebe las funcionalidades de cada comando ( para el comando **ethool** se usó el comando **ifconf** en reemplazo)





* 1. Construya un Shell que permita saber los procesos que están corriendo en un momento dado en un servidor. Muestre el nombre del proceso, su identificador, % de memoria y % de CPU utilizada.



PID: Representa la ID del proceso

COMMAND: Indica el nombre del comando con el que está ejecutando el proceso

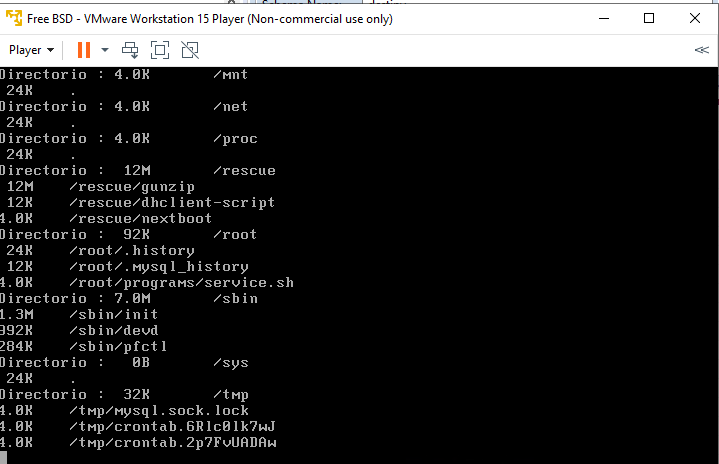
%CPU: Indica cantidad de CPU utilizada

%MEM: Indica la cantidad de memoria usada en el proceso.

* 1. Cree un Shell que recorra todos el file system y muestre el tamaño total de los directorios ubicados en la raíz del sistema y los 3 archivos más grandes (en bytes) dentro de cada uno

Primero creamos un script que nos muestre la información que hay en cada uno de las carpetas ubicadas en la carpeta raiz del sistema

Luego verificamos que correctamente muestre los 3 archivos más grandes en bytes dentro de cada uno



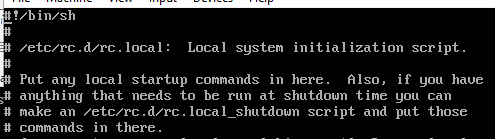
* 1. Para los grupos de 3 personas, realice scripts similares a los indicados en los numerales anteriores en poweShell.

## Otras configuraciones de motores de bases de datos

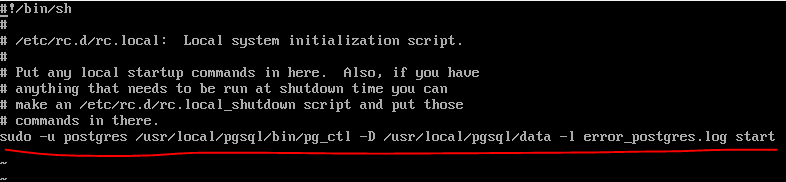
* 1. En los servidores en los que instaló las bases de datos, configure el sistema operativo de tal manera los motores de base de datos suban automáticamente cuando arranque el sistema operativo.

POSTGRESQL – Linux SLACKWARE

Lo primero que debemos hacer es modificar el archivo que inicia todas las configuraciones al bootear el sistema operativo. Este archivo lo encontramos en la ruta “cd etc/rc.d” y una vez dentro del directorio, editamos el archivo con el comando “vi rc.local” y nos debe aparecer lo siguiente:



Podemos notar que nos muestra un mensaje que dice que pongamos todos los comandos que necesitamos que se corran cuando la maquina este iniciando. Así que es justo lo que haremos, digitaremos el comando que permite inicializar el servidor de la base de datos

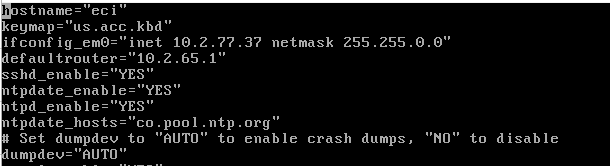


Guardamos el archivo y apagamos la máquina. Nuevamente la iniciamos y cuando nuestro servidor este iniciado, nos aparecerá el siguiente mensaje

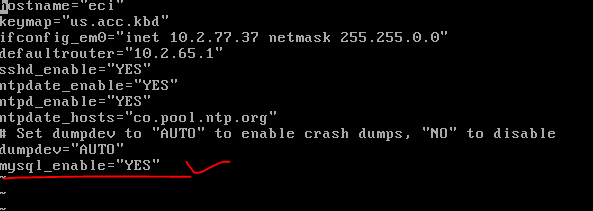


MARIA DB – Linux FREEBSD

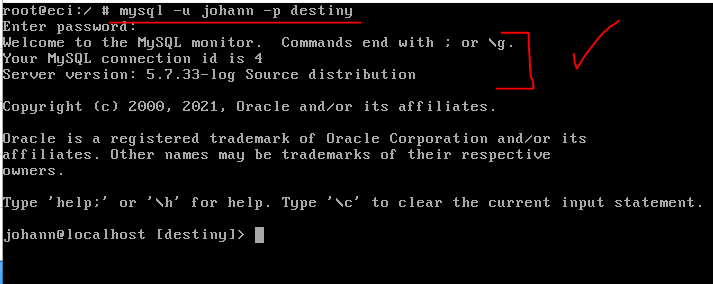
En Linux FreeBSD es mucho mas sencillo auto iniciar el servidor de nuestras bases de datos, para ello iremos al directorio “cd /etc” y utilizando el editor vi, editaremos el archivo rc.conf con el siguiente comando “vi rc.conf” y nos debería mostrar algo como esto:



Luego, para poder iniciar nuestro servidor de mysql después de iniciar nuestro sistema operativo, debemos agregar la siguiente línea: mysql\_enabled=”YES”, con esto estamos diciéndole que inicie mysql con los archivos de configuración del sistema operativo



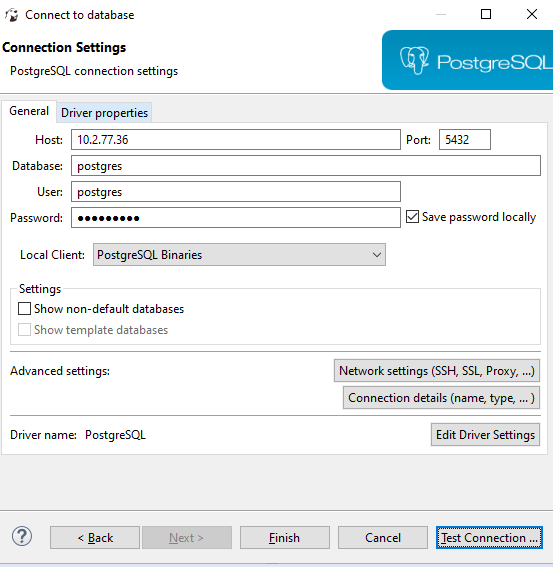
Guardamos el archivo y podemos testear reiniciando la maquina y accediendo directamente a una base de datos con el comando “mysql -u Johann -p destiny” lo cual nos debería permitir acceder a la base de datos sin errores de conexión al servidor o cualquier otro



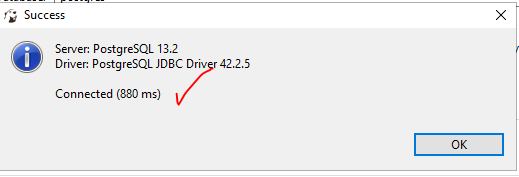
* 1. Usando un cliente de conexión a motores de base de datos, conéctese a sus bases de datos desde una máquina remota y vea el contenido de las tablas.

Conexión remota POSTGRESQL

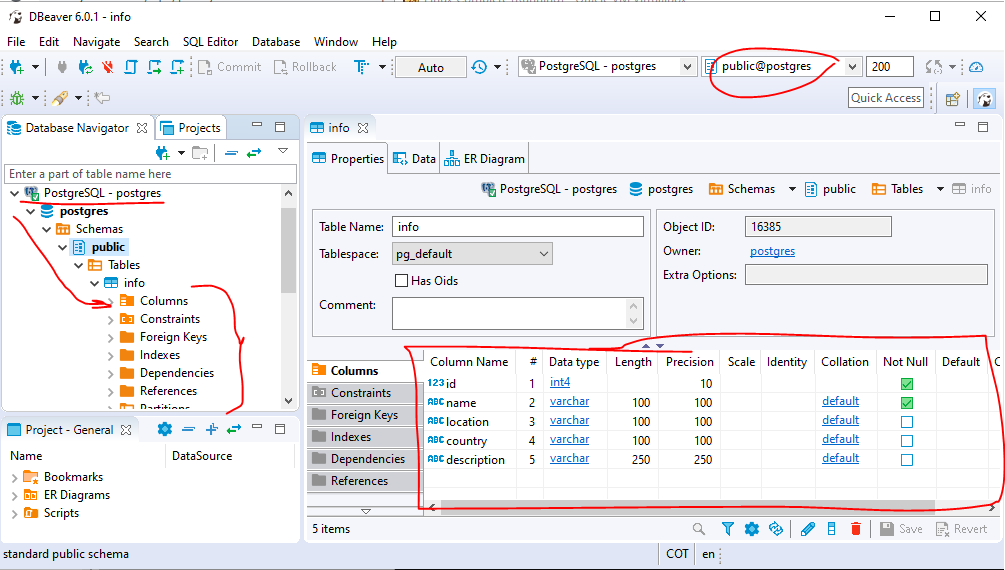
Lo primero que haremos será probar nuestra conexión con ayuda de un software llamad DBeare, e ingresmos los datos de la base, con el puerto 5432 que es el puerto por el cual se conecta PostgreSQL, y entramos con un usuario llamado postgres a la base de datos postgres



Si nuestra base de datos esta condifurada para conexión remota desde la ip desde la cual nos vamos a conectar, entonces debería conectarse de manera exitosa



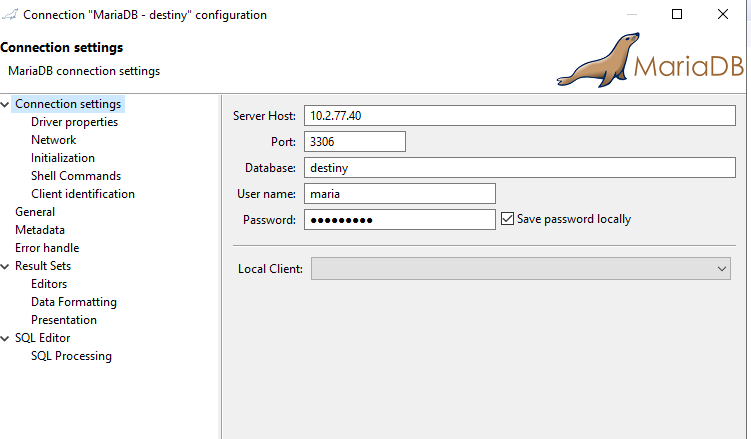
Para poder ver el contenido de las tablas, usaremos el software por el cual nos conectamos, y hacemos conexión con la base de datos de la cual queremos obtener información. Sabremos que estamos conectados ya que nos aparecerá un check verde en el gestor de bases de datos, y si desplegamos la información, en la sección de tablas, encontraremos la lista de cada una y sus propiedades, como las columnas, llaves, triggers y demás. Y en la parte de la derecha, encontraremos la información almacenada en la tabla.

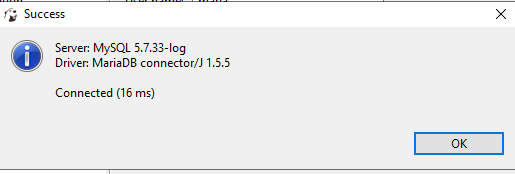


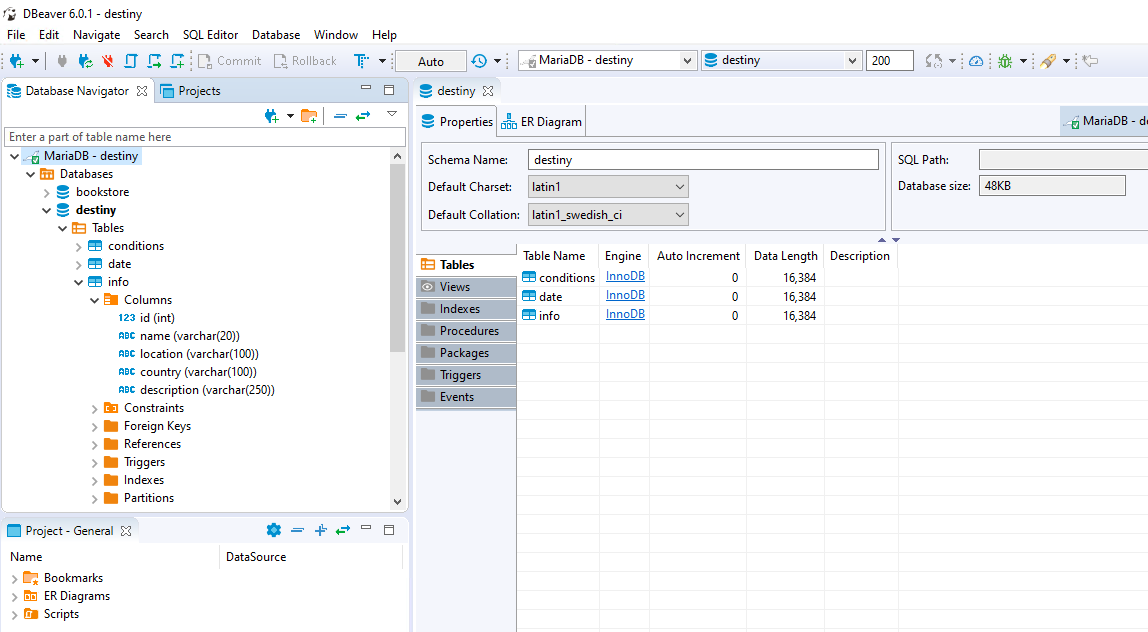
Conexión remota MARIADB

En este punto lo primero que vamos hacer es probar la conexión de la base de datos con ayuda de DBeare , para ello , probamos la conexión con el Puerto 3306 Sql y entramos con usuario llamado maria a la base de datos ‘destiny’





Si la base de datos esta configurada para el acceso remoto desde la ip a la cual nos vamos a conectar debería mostrar un mensaje de conexión exitosa 



En este punto ya podemos editar , configurar la base de datos y ver toda la información relacionada a la base de datos tanto como tablas y los atributos en ellas.

CONCLUSIONES

* Se puede concluir que la importancia de los motores de bases de datos son parte fundamental dentro de la infraestructura computacional de una organización aparte de los sistemas operativos.
* Se logró afianzar conocimientos sobre el funcionamiento de la conexión de las bases de datos y de cómo podemos acceder remotamente a ellas al igual que podemos hacer uso de comandos en Shell para conocer información relevante de los servidores.
* Dimensionar la importancia de las bases de datos también radica en comprender su papel en el almacenamiento de datos estructurados, aplicativos, así como también en las distintas operaciones soportadas por una organización.

BIBLIOGRAFIA

* <https://www.tecmint.com/install-apache-mariadb-and-php-famp-stack-on-freebsd/>
* <http://minubeinformatica.com/cursos/montaje-de-un-servidor-web/un-poco-de-comandos-freebsd-antes-de-continuar-comandos-linux-basicos>
* <https://www.freebsd.org/cgi/man.cgi?xargs>
* <https://www.freebsd.org/cgi/man.cgi?du>
* <https://www.freebsd.org/cgi/man.cgi?top(1)>
* <https://www.freebsd.org/cgi/man.cgi?crontab%285%29>
* <https://www.freebsd.org/cgi/man.cgi?ifconfig(8)>
* <https://www.freebsd.org/cgi/man.cgi?route>
* <https://www.freebsd.org/cgi/man.cgi?query=netstat&sektion=1>
* <https://www.freebsd.org/cgi/man.cgi?query=vnstat&sektion=1&manpath=freebsd-release-ports>
* <https://www.cyberciti.biz/tips/freebsd-timeclock-synchronization-with-ntp-server.html>